

Amendments to the Claims:

51. (Previously presented) A method for pushing information to a client in an information processing system, the method comprising the steps of:

providing a platform implementing the information processing system and a system client for practicing the method;

gathering facts concerning user activity;

forming the gathered facts into fact nets;

storing the fact nets within the platform-implemented system;

identifying fact nets defining evolving probable user interests; and

pushing selected facts from the identified fact nets to the system client.

52. (Previously presented) The method as set forth in Claim 51, including the steps of deriving new facts from the facts within the fact nets, and incorporating the new facts into the fact nets.

53. (Previously presented) The method as set forth in Claim 51, including the step of repeating all steps until stopped.

54. (Previously presented) The method as set forth in Claim 53, wherein the repeating step includes executing the previous steps concurrently as independent threads.

55. (Previously presented) The method as set forth in Claim 54, including the steps of defining a no-longer-valid fact, and pruning no-longer-valid facts from the fact nets.

56. (Previously presented) The method as set forth in Claim 52, including a step of triggering the deriving and the pushing steps upon the formation of the fact nets.

57. (Previously presented) The method as set forth in Claim 56, wherein the triggering

occurs upon the derivation of new facts, thereby defining an iterative process of reevaluation and reporting.

58. (Previously presented) The method as set forth in Claim 57, including a step of deriving every possible fact that can be derived each time a new fact is added.

59. (Previously presented) The method as set forth in Claim 51, wherein the client is an application program that displays facts for user viewing.

60. (Previously presented) The method as set forth in Claim 51, wherein the client is an application program that archives facts for later use.

61. (Withdrawn) A system for monitoring events in an environment, for making inferences about the monitored events, and for reporting selected inferences to a client, the system comprising:

at least one observer agent for monitoring a selected event of an environment, and for creating a primitive fact which incorporates a status of the monitored event;

a dynamic user model for storing created facts, the stored facts being accessible by the agents;

at least one reporter agent for examining created facts, for defining and identifying reportable facts, and for delivering a copy of the reportable facts to a receiving client; and

the agents, the dynamic user model, and the client being implemented within a single platform.

62. (Withdrawn—currently amended) The system as set forth in Claim [[16]] 61, wherein the at least one observer agent, and the at least one reporter agent each defines an

independent, concurrent programming thread.

63. (Withdrawn—currently amended) The system as set forth in Claim [[17]] 62, wherein the at least one reporter thread is triggered by the creation of each new primitive fact, whereby delivery of reportable facts to a client is triggered by occurrence of monitored events.

64. (Withdrawn) A method for creating and using a dynamic user model to push information to a client in an information processing system, the method comprising the steps of:

providing a single platform for the practice of the method;

building and maintaining a dynamic user model within the platform;

implementing a receiving client within the platform; and

using the dynamic user model for pushing information to the receiving client by performing the following steps,

examining the contents of the dynamic user model to identify fact nets defining evolving

probable user interests, and

pushing selected facts from the identified fact nets to the receiving client.

65. (Withdrawn—currently amended) The method as set forth in Claim [[19]] 64, further including the step of triggering the information pushing on changes in the dynamic user model.

66. (Withdrawn—currently amended) The method as set forth in Claim [[19]] 64, wherein the dynamic user model includes information defined by a current web page being viewed by a user of the information processing system, and wherein the web page information in the dynamic user model is used to access and push news stories related to the web page, and to access and push stock prices related to the web page.

67. (Withdrawn—currently amended) The method as set forth in Claim [[19]] 64, wherein the dynamic user model includes information defined by keystrokes made by a user of the information processing system, and wherein the keystroke information further defines correct and incorrect user keystrokes, and the keystroke information is used to construct typing exercises aimed at improving user typing skill, and wherein the exercises are pushed to the user.

68. (Withdrawn—currently amended) The method as set forth in Claim [[19]] 64, wherein the dynamic user model includes information derived from user interactions with an application program of the information processing system, and the application program interaction information is used to access additional information about the application program via a network, such as one of program updates and related and competing programs.

69. (Previously presented) A system for pushing information to a client in an information processing system, the system comprising:

means for gathering facts concerning user activity and for forming the gathered facts into fact nets;

means for identifying fact nets defining evolving probable user interests;

means for pushing selected facts from the identified fact nets to a system client; and

single platform means implementing the system and the client.

70. (Previously presented) The system as set forth in Claim 69, including means for deriving new facts from the facts within the fact nets, and for incorporating the new facts into the fact nets.

71. (Previously presented) The system as set forth in Claim 69, including means for

continuing until stopped.

72. (Previously presented) The system as set forth in Claim 71, wherein said continuing means further includes each of said previous means defining concurrent, independent program threads.

73. (Previously presented) The system as set forth in Claim 72, further including means for defining a no-longer-valid fact, and for pruning no-longer-valid facts from the fact nets.

74. (Previously presented) The system as set forth in Claim 70, including means for triggering the deriving means and the pushing means upon the formation of the fact nets.

75. (Previously presented) The system as set forth in Claim 74, wherein the triggering occurs upon the derivation of new facts, thereby defining an iterative means of reevaluation and reporting.

76. (Previously presented) The system as set forth in Claim 75, including means for deriving every possible fact that can be derived each time a new fact is added.

77. (Previously presented) The system as set forth in Claim 69, wherein the client is an application program that displays facts for user viewing.

78. (Previously presented) The system as set forth in Claim 69, wherein the client is an application program that archives facts for later use.

79. (Previously presented) A system for monitoring events in an environment, for making inferences about the monitored events, and for reporting selected inferences to a client, the system comprising:

at least one observer agent for monitoring a selected event of an environment, and for creating a primitive fact which incorporates a status of the monitored event;

AM9-97-133-US2

-6-

09/864,537

a dynamic user model for storing created facts, the stored facts being accessible by the agents; and

at least one reporter agent for examining created facts, for defining and identifying reportable facts, and for delivering a copy of the reportable facts to a receiving client.

80. (Previously presented) The system as set forth in Claim 79, wherein the at least one observer agent, and the at least one reporter agent each defines an independent, concurrent programming thread.

81. (Previously presented) The system as set forth in Claim 80, wherein the at least one reporter thread is triggered by the creation of each new primitive fact, whereby delivery of reportable facts to a client is triggered by occurrence of monitored events.

82. (Previously presented) The system as set forth in Claim 79, including at least one fact deriving agent for examining all existing facts, for creating new facts from one or more existing facts, and for linking each new fact to a parent fact of the new fact, forming fact nets of linked facts, and further including all fact nets being stored in the dynamic user model.

83. (Previously presented) The system as set forth in Claim 82, including a fact pruning agent defining a no-longer-valid fact and a descendant of a fact, the fact pruning agent eliminating each no-longer-valid fact and all its descendants from the fact nets.